ChinaXiv合作期刊

Spark 环境下 K-means 初始中心点优化研究综述 *

行艳妮，钱育蓉，南方哲，赵京霞
(新疆大学 软件学院，乌鲁木齐 830046)

摘　要：为了能够及时了解 Spark 环境下经典聚类算法 K-means 的最新研究进展，把握 K-means 算法当前的研究热点和方向，针对 K-means 算法的初始中心点优化研究进行综述。首先介绍了内存计算框架 Spark 和 K-means 算法，并分析了 K-means 算法聚类不稳定性的成因和影响，其目的在于指出优化 K-means 算法的重要性。详细介绍了目前在 Spark 环境下优化 K-means 初始中心点的主要方法和最新研究现状，并展望了 K-means 初始中心点优化问题的未来研究方向。

关键词：K 均值算法；分布式内存计算框架；算法优化；聚类算法
中图分类号：TP391　　doi: 10.3969/j.issn.1001-3695.2018.10.0609

Survey of optimization on K-means algorithm in Spark

Xing Yanni, Qian Yurong, Nan Fangzhe, Zhao Jingxia
(*College of Software*, *Xinjiang University*, *Urumqi* 830046, *China*)

Abstract: In order to understand the latest research progress of the classical clustering algorithm K-means in Spark environment, and grasp the current research hotspots and directions of K-means algorithm, this paper reviews the initial center point optimization research on K-means algorithm. Firstly, it introduces the memory computing framework Spark and K-means algorithms, and analyzes the cause and effects of clustering instability of K-means algorithm, which aimed to point out the importance of optimizing K-means algorithm. And it introduces the main methods and the latest research status of optimizing the initial center point of K-means in Spark environment in detail, and also discusses the future research trends in initial center point optimization of K-means.

Key words: K-means; distributed memory computing framework; algorithm optimization; clustering algorithm

0　引言

K-means 作为机器学习[1]迭代聚类算法中的一种经典简单的算法，在数据挖掘[2]和模式识别方面得到了广泛应用。由于 K-means 算法运行时初始聚类中心点和簇数的不确定性，使聚类结果存在着很大的不稳定性，以致于算法处理数据时会出现准确率降低、收敛速度慢等问题，特别是近年来，随着全球计算机和信息技术的迅猛发展，K-means 在处理规模日益增长的数据[3]时，聚类效果不佳和算法效率低下等缺点表现得更加明显，无法满足实际大数据场景下应用需求。因此在大数据环境下，针对 K-means 算法的初始聚类中心点优化问题，很多学者首先在 K-means 算法中使用各类改进算法优化其中心点选取过程，以减少算法迭代次数，提高收敛速度和准确性；同时基于 Spark[4,5]内存计算框架，进一步提高了算法处理大规模数据的效率以及对大数据的适应能力。本文详细介绍了 K-means 算法的几种改进方法，以及其在 Spark 环境下的并行化现状。

1　Spark 内存计算框架

Spark 是由 Berkeley 的 AMPLab 于 2009 年提出的一种由 Scala 语言实现的大数据计算框架，既兼容了 Hadoop[6]中 MapReduce[7]的可扩展性和容错性等优点，同时引入了内存计算的概念。Spark 采用弹性分布式数据集[5]（resilient distributed datasets，RDD）数据结构将算法的中间结果保存在内存中，使其更适用于反复迭代运行的应用程序，如交互式数据挖掘和机器学习算法；并通过数据集血统（lineage）[5,8]和检查点机制[9]实现了系统容错，解决了迭代算法在 Hadoop 下不断进行磁盘访问而造成的性能损失问题。由于 Spark 能够部署在通用平台上，并具有可靠性（reliable）、可扩展性（scalable）、高效性（efficient）、低成本（economical）等优点[10]，目前已被广泛应用于大规模数据处理过程。

为支持不同应用场景下的大数据处理，Spark 已经发展成为包含众多子项目的大数据计算平台，如图 1 所示，包括了基于 Spark 核心组件的可扩展机器学习函数库 MLlib[11-12]，实现了基于 RDD 的 K-means 算法，并且初始中心选取采用了 K-means 算法[13]，然而 MLlib 中的 K-means 算法核心没有改变，在执行过程中，仍存在大量不必要的距离计算，影响算法的运算效率。因此基于 Spark 的 K-means 算法并行化[14-15]研究仍在进行中。

图 1 中 Spark Core 是 Spark 生态系统的核心，Spark SQL[16]和 Shark[17]支持结构化数据 SQL 查询与分析的查询引擎，MLbase 提供了机器学习功能的系统，MLlib 为底层的分布式机器学习库，还有并行图计算框架 GraphX[18]、流计算框架 Spark Streaming[19]、内存分布式文件系统 Tachyon 及资源管

收稿日期：2018-10-31；**修回日期：**2018-11-23　**基金项目：**国家自然科学基金资助项目（61562086，61462079）；新疆维吾尔自治区教育厅创新团队资助项目（XJEDU2016S035）

作者简介：行艳妮（1994-），女，陕西渭南人，硕士研究生，主要研究方向为数据挖掘（1377429024@qq.com）；钱育蓉（1980-），女，教授，博士，主要研究方向为网络计算和遥感图像处理；南方哲（1994-），女，硕士研究生，主要研究方向为计算机视觉和单图像超分辨率重建等；赵京霞（1995-），女，硕士研究生，主要研究方向为图像处理与模式识别.

chinaXiv:201901.00067v1

ChinaXiv合作期刊

chinaXiv:201901.00067v1

理框架 MESOS[20]等子项目，这些子项目在 Spark 上层提供了更高层、更丰富的计算范式。



图 1　Spark 生态系统

Fig. 1　Spark ecosystem BDAS

并行计算是指同时使用多个计算资源来解决一个大型复杂的计算问题的方式。在 Spark 中实现算法并行化[18,21]过程如图 2 所示，首先通过将一个任务分解为多个子任务，然后将其分配给不同的处理节点，各个处理节点之间相互协同，并行地执行子任务，最终将所有子任务的结果合并为最终结果输出。

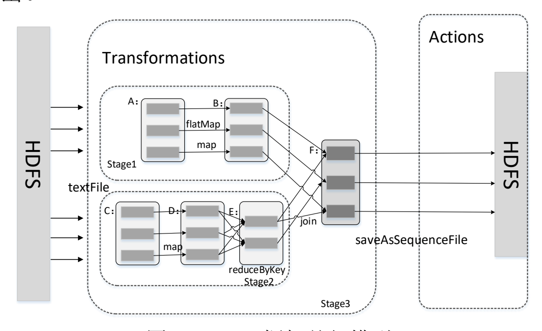


图 2　Spark 框架并行模型

Fig. 2　Spark parallel model

2　K-means 算法概述

2.1　K-means 算法

K-means 算法[22]（也称为 K-均值）是 1967 年由 MacQueen 首次提出的，如今它已成为一种经典聚类算法[23]，被广泛应用于数据挖掘和模式识别中。K-means 算法是一种无监督学习，同时也是一种基于划分的聚类算法，即需预先指定聚类数以及中心，不断更新聚类中心，逐步降低目标函数的误差值，当目标函数值收敛时，得到最终聚类结果。

K-means 是通过度量数据间的距离对数据进行分类的一种算法，最终使得相似程度高的数据都处在同一个簇内，相似程度低的数据则处于不同的簇中，算法步骤如下所示：

a)从给定数据集中随机选取 k 个数据对象作为初始聚类中心 $C(C_1,C_2,...,C_k)$；

b)对于数据集中的每个数据 X_i，计算其到初始聚类中心 C 的距离，找到离它最近的初始聚类中心 C_j，并将其分配给 C_j 所标明的类别；

c)更新每个类别的中心 C_j 将其作为该类别的所有样本的均值；

d)不断迭代以上过程直到每个中心不再变化或达到迭代次数为止。

在 K-means 算法过程中，数据之间的相似程度常由欧几里德距离来表示。如在多维空间中，每个数据都是 n 维向量，空间中数据点 X 和中心点 C 之间的欧式距离则定义为

$$\mathrm{dist}(x_i,c_i)=\sqrt{\sum_{i=1}^{n}(x_i-c_i)^2} \tag{1}$$

其中：n 是维数，x_i,c_i 分别是 X 和 C 的第 i 个属性值。

另外，通常使用 SSE（sum of squares due to error，误差平方和）来度量聚类结果质量，是 K-means 算法的目标函数为

$$SSE=\sum_{j}^{n}\sum_{i=1}^{k}\sum_{x\in S_i}(d(c_i,x_i))^2 \tag{2}$$

其中：d 是欧氏空间中两个点的欧氏距离，k 是簇的个数，x 是数据点，c_i 是第 i 个簇的中心，S_i 表示第 i 个簇中数据点的集合。在 K-means 算法运行过程中，由于目标函数的非凸性[24]，初始聚类中心点的随机选取会形成许多不同的局部最优解，无法达到全局最优，使得算法更易陷入局部最优[25]。

K-means 算法的计算复杂度为 $O(tKmn)$，其中 t、K、n、m 分别指迭代次数、簇类个数、特征维数和待分类的数据规模，通常，$t,K,n \ll m$，由此可见处理大规模数据集时，K-means 时间复杂度接近于线性。

2.2　K-means 算法改进

由上述可知，K-means 算法原理较为简单，收敛速度快[26]，然而在实际应用时，K-means 算法表现出了一些局限性，在运行时需指定初始参数，如簇类个数 k、初始聚类中心等，这些参数初始值的设置对聚类效果影响巨大。

针对这些局限性，国内外学者提出了许多改进 K-means 算法的策略，如使用网格优化[27]、残差分析[28]、Hash 算法[29]、萤火虫优化[30]、基于密度和距离改进[31]等优化方法预先确定初始聚类中心点，解决 K-means 随机选取初始中心点带来的不稳定性问题；使用余弦相似度度量数据间距离[32]、基于方差的聚类评估标准[33]等方法改善 K-means 算法的聚类效果，提高了 K-means 算法的准确性和稳定性；还有研究提出了一种精确减少采样 K-means 算法[34]，可有效处理不平衡数据集；然而这些研究仅针对 K-means 算法本身进行优化，未借鉴当前流行的大数据计算框架的优势，难以满足对大规模数据高效聚类的需求。

近几年流行的大数据计算框架有 Hadoop、Spark、Flink[35]等，现有研究如文献[36,37]在 Hadoop 下并行化实现了 K-means 算法，文献[38]在 Flink 上改进了 K-means 算法，Kusuma I[39]结合 Spark 的 RDD 数据结构实现了 K-means 对大规模数据的聚类，或将 K-means 聚类分析的过程直接放在 Spark 环境下进行[40,41]，这些研究都提高了 K-means 处理大数据数据的效率；也有不少学者利用 K-means 算法简单易实现的优点对比了大数据计算框架的优势[42,43]，证明了 Spark 平台在机器学习迭代式计算方面的优势，因此，将改进后的 K-means 算法应用到 Spark 中并结合其内存计算优势，是未来很有价值的一个研究方向。本文主要总结了在 Spark 环境下针对 K-means 算法初始中心点的改进研究现状。

3　基于 Spark 的 K-means 算法优化策略

大数据环境下，K-means 算法随机初始中心点会对聚类结果的准确性以及算法的收敛速度产生较大的影响，因此，针对随机初始中心点产生的不稳定性问题，目前有大量研究尝试解决该问题。

首先大多数研究人员在 Spark 下并行化 K-means 算法，其基本过程如图 3 所示，分为 map、combine 和 ReduceByKey 三个阶段。

ChinaXiv合作期刊

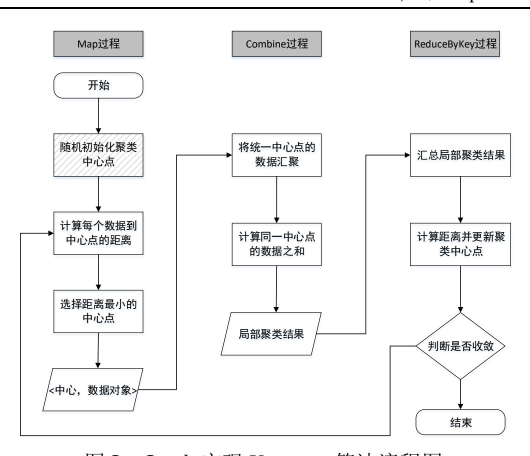


图 3　Spark 实现 K-means 算法流程图

Fig. 3　Flow chart of K-means Algorithm on Spark

基于上述基础，研究人员对 K-means 算法本身的原理进行优化，从而在提高算法运行效率的同时提高准确性，主要通过使用对数据集划分的方法、预聚类初始中心点的方法和引入全局优化思想的方法等策略来优化 K-means 算法初始中心点的过程，下面将详细介绍着三种优化方法。

3.1　基于数据划分的 K-means 改进策略

K-means 算法在 Spark 下并行时，若直接读取整个数据集并转换成 RDD 格式，可能会由于数据集构成的 RDD 过大无法充分利用 Spark 平台的内存计算特性，因此有必要对数据进行划分。

但若随机划分数据集，可能会增加算法的迭代次数对算法效率影响颇大，有研究人员使用二叉树[44]、四叉树[45]、将数据空间划分为单元格[46]等对数据集进行划分，发现使用二叉树划分有可能会使划分后的数据集与原数据集的分布不同，影响并行化算法聚类效果和效率，由于 Kd 树(K-demension tree)[47]可以按数据集分布情况划分数据，对 K-means 初始中心点选择和并行化都有很大益处，因而被广泛应用。

Kd 树是一种用于分割多维数据空间的数据结构，其中 k 指数据空间的维数，其构建的基本思想为[48]：根据数据分布密度情况，分别计算数据每个维度上的方差值，同时考虑左右子树的平衡性，选取分割维度数据的中值进行分割，不断迭代进行分割过程直到算法满足迭代次数为止。

Kd 树构建过程在低维数据中表现良好，但面对大规模数据，由于其维度的增加，会出现算法性能急剧下降的现象。Spark 内存计算框架通过分解 Kd 树构建过程，将任务分配到多个计算节点上进行，充分利用 Spark 中每个节点的计算能力共同构建 Kd 树，有效缓解了算法性能下降问题，减少了 Kd 树构建的时间消耗，进一步提高 Kd 树对大数据的适应性。

基于上述研究，考虑到 Kd 树多维空间数据的分割以及搜索方面所表现出的优势，可用来改善 K-means 原始随机选取初始中心点的缺陷，改进后的 K-means 算法流程如图 4 所示，使用 Kd 树对数据集分区帮助 K-means 算法准确地选取初始中心点，避免了随机选取出现孤立点数据或中心点过于密集的问题。

当前已有的研究如文献[49,50]首先使用 Kd 树判断数据集各个区域的数据密度，为 K-means 算法提供选取初始中心点的依据，并通过实验证明 Kd 树数据结构对 K-means 算法在运行时间和聚类准确性的有效提高；然而 Kd 树本身较慢的建树速度随着数据量的增大会进入瓶颈期，因此其建树效率还待进一步提高。为高效准确对大规模数据进行聚类，马菁等人[51]实现了在 Spark 环境下的 Kd 树并行预处理的 K-means 算法，充分继承了 Spark 的内存计算和良好容错性优点，同时提高了算法运行效率，特别是在大数据集上表现良好。

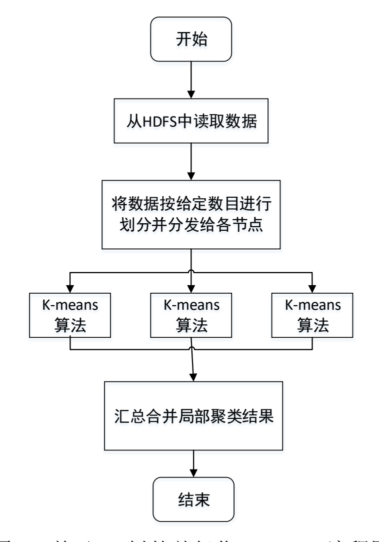


图 4　基于 kd 树的并行化 K-means 流程图

Fig. 4　Flow chart of parallelized K-means on Kd-tree

3.2　基于预聚类算法的 K-means 算法改进

K-means 算法过度的依赖初始中心点，很容易导致聚类结果陷入局部最优，在面对大规模数据时，此缺陷更加明显并且会出现算法效率低下的问题，因此，一部分学者采用预聚类、预抽样以及最大最小距离等方法预先选取 K 个聚类中心，作为 K-means 算法的一部分输入，改变 K-means 算法随机选取初始中心点的过程，达到提升算法稳定性的目的。下面主要介绍预聚类 Canopy 算法和最大最小距离算法对 K-means 算法的改进。

3.3　基于 Canopy 预聚类算法的改进方法

Canopy 算法是由 McCallum[52]在 2000 年提出的无监督预聚类算法，适用于高维度的大数据预聚类分析，通常用做 K-means 算法预处理步骤，避免其初始中心点选取的盲目性。Canopy 算法可以通过简单的距离计算方法将初始数据集划分为多个不完全重叠的子集 Canopy，很好地剔除数据集中的噪声点或孤立点，其步骤[53]如下：

a)如图 5 所示，初始距离阈值为 T_1, T_2 （$T_1 > T_2$），将给定数据集的数据存入 list 集（候选中心点）中，并从 list 中随机选择数据点 C_1，由式（1）计算 List 中其他数据点 x_i 与 C_1 之间的距离 d_{i1}；

b)若 $d_{i1} < T_1$，则将其划分到一个 canopy 中，且若 $d_{i1} < T_2$，则将数据点 x_i 从 list 中移除，对于满足 $T_2 < d_{i1} < T_1$ 的数据，重新计算出新的中心点并根据数据与新中心点之间的距离重新划分该数据所属的类；

c)不断循环以上操作，直到 list 集中元素为空，算法结束。

Canopy 算法每次仅比较同一区域内数据与中心点之间的距离，减少了比较次数从而大大降低了聚类的运行时间，可提高算法的计算效率，然而近年来数据的爆炸式增加，预聚类 Canopy 算法的距离计算次数和比较次数直线上升，导致算法运行速度无法满足实时性需求。

为了提高 K-means 算法的稳定性和准确性，解决初始 K 值和初始中心点的问题，Zhang 等人[53]将一种具有密度参数的 Canopy 算法用做 K-means 的预处理过程，其输出结果作为 K-means 算法的簇数和初始中心点，经实验证明基于 Canopy 的 K-means 算法相较于传统 K-means 算法具有更高

chinaXiv:201901.00067v1

ChinaXiv合作期刊

的聚类精度，并且对于噪声数据有了很好的处理能力；徐鹏程等人[54]在 Spark 并行计算框架下，引入 Canopy 算法消除了 K-means 算法初始 K 值的不确定性，并通过最大最小距离算法优化了 K-means 算法中初始中心点的选择过程，验证了改进后的 K-means 算法在大数据环境下的稳定性、准确性以及计算效率。

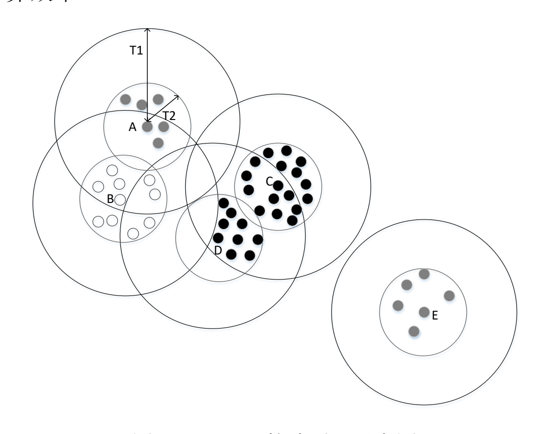


图 5　Canopy 算法原理示意图

Fig. 5　Schematic diagram of Canopy algorithm

3.4　最大最小距离算法

然而，Canopy 算法在实际应用中需预先设置距离阈值 T_1, T_2 等初始值，具有很大的随机性，对算法最终结果的准确性和效率都有着巨大的影响，为更好地应用 Canopy 算法，很多学者将最大最小距离算法与 Canopy 算法进行结合，对 Canopy 中心点的选取和设置进行了优化，取得了不错的结果。

最大最小距离算法（the maximum and minimum distance method）是 Katsavounidi 等人[55]于 1994 年提出的针对聚类算法中初始聚类中心点的选择方法之一，其核心思想是尽可能选择距离已有中心点较远的点来避免选取中心点不当的问题，算法步骤如下：

a)从包含 N 个数据的数据集中任一选取一个数据点作为第一个聚类中心点 C_1，并选取距离 C_1 最远的数据点作为第二个聚类中心点 C_2，两点之间距离计算采用式（1）；

b)计算各个样本到各个聚类中心的距离 d_{ij}，并判断式（3）是否成立

$$d_j = \max[\min[d_{i1}, d_{i2}, ..., d_{ik}]] > \theta * \| c_1 - c_2 \| \tag{3}$$

其中 θ 为比例系数，若成立，则 $c_i = x_j$，即 x_j 成为聚类中心，不断循环此步骤，直到不再有新的聚类中心出现；

c)根据上述得到的聚类中心 $\{c_1, c_2, ..., c_k\}$，将数据集中的数据点分到距离最近的类（中心点）中去。

然而最大最小距离算法在面对大规模数据时会出现中心点选取过程较为耗时的问题，大大影响了算法的执行效率，为此在 K-means 算法运行时充分借鉴 Spark 内存计算的优势，提升算法的执行效率。

众多学者将最大最小距离算法运用到 K-means 算法的初始过程中，如文献[56]实现了基于最大最小距离算法的 K-means 算法，应用最大最小距离对数据集进行预处理，输出 K 个中心点作为 K-means 的输入，提高了 K-means 算法的性能，并且改进了最大最小距离算法不能处理噪声数据的问题；Mao 等人[57]通过引入最大最小距离算法对 Canopy 算法中心点的选取与设置进行优化，并为适应大数据的应用场景，在 MapReduce 并行计算框架对算法进行了并行加速，实验表明，该算法在时间、准确性等方面都有明显提高；文献[54]在 Spark 并行计算框架下，通过最大最小距离算法优化了 K-means 算法中初始中心点的选择过程，提高了 K-means 算法在大数据环境下的稳定性、准确性以及计算效率。

3.5　基于元启发式的 K-means 算法改进

近年来，元启发式优化算法，如遗传算法（Genetic Algorithm，GA）[58]、禁忌搜索算法（tabu search，TS）[59]、粒子群优化算法（Particle Swarm Optimization，PSO）[60-61]等，由于其通用性、简单性、鲁棒性强以及适合并行处理等优点得到了广泛应用。将启发式算法的思想贯穿到 K-means 算法的整个过程中，解决了 K-means 算法易陷入局部最优的缺陷，减少了对初始中心点的依赖，提高了聚类结果的稳定性和准确性。下面详细介绍基于 Tabu Search 和粒子群优化的 K-means 算法改进方法。

3.5.1 禁忌搜索算法

禁忌搜索算法是一种用来跳脱局部最优解找到全局最优解的搜索方法，它主要思想是使用禁忌列表限制数据选择次数，从而避免算法陷入局部最优。禁忌搜索算法的流程如图 6 所示，从一个随机的初始中心点出发，根据指定的搜索策略以及各个参数，在当前初始中心点的邻域内寻找新的聚类中心点，不断寻找使得聚类结果最优的聚类中心，直到当前聚类中心不再改变或到达最大迭代次数为止。

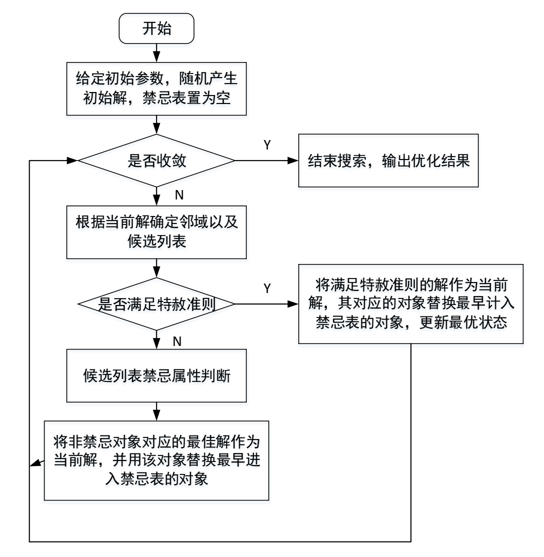


图 6　禁忌搜索算法流程图

Fig. 6　Flow chart of Tabu search algorithm

禁忌搜索算法优化能力较好，但也存在计算复杂度高、耗时较长的问题，为适用于当前大数据应用的需求，在 Spark 框架下对算法的两个过程进行了并行，分别是数据分配给最近中心点的迭代过程和更新中心点的过程，大大提高了算法运行效率和可扩展性；同时将禁忌搜索算法的优化思想应用到 K-means 算法中，通过禁忌列表限制算法频繁选择同一数据点而避免其陷入局部最优的情况，再通过邻域和候选表设计，使得算法的初始中心点选择范围更为全面，实现对 K-means 算法的聚类精度和稳定性的提高。

目前，文献[62,63]通过使用禁忌搜索算法对 K-means 算法初始中心点进行优化，相较于传统 K-means 提高了聚类结果的准确度并克服了 K-means 算法易陷入局部最优的缺陷，但在计算时间等方面改进不大，Lu 等人[64]在 Spark 环境下引入禁忌搜索算法，并将最大最小距离算法与禁忌搜索算法相结合，优化初始中心点的同时提高了算法的运行效率。

3.5.2 粒子群优化

粒子群优化算法是由 Kennedy 和 Eberhart 通过研究鸟群觅食过程而提出的一种群体智能优化算法，其基本思想[65]

chinaXiv:201901.00067v1

ChinaXiv合作期刊

chinaXiv:201901.00067v1

是借鉴群体中个体之间的协作和信息共享来寻找问题的最优解，首先初始化一群随机粒子（即问题的解），同时初始化每个粒子的速度、位置等参数值，每个粒子通过不断迭代更新自身的速度和位置等找到最优解。

虽然粒子群优化算法全局优化能力较强，然而处理现阶段的大数据规模时仍显得力不从心，为提高算法处理大规模数据的能力，依托 Spark 分布式计算框架，结合 Spark 的 RDD 缓存等特性，多个节点并行迭代更新初始解，提高算法的运行效率；同时将粒子群优化算法引入到 K-means 算法中优化 K-means 算法的整个过程，有效提高了 K-means 算法的收敛速度和聚类结果。

当前许多学者已经基于这些方面进行了深入研究，如 Lin 等人 [66]通过结合粒子群优化和多类合并改进 K-means 算法，利用粒子群算法寻找初始聚类中心，加速了 K-means 算法的聚类速度，并进行多类合并操作，实现了聚类算法的最优划分。许明杰等人[67]在 Spark 环境下，利用粒子群优化算法来提高 K-means 的全局搜索能力，得到初始聚类中心，并充分利用 Spark 的迭代计算能力将其并行化，提升了 K-means 对大数据的处理速度，缩短了算法的整体运行时间。然而，PSO 算法也存在着局部搜索能力较差、搜索精度不高且易陷入局部极值等缺点[60]，刘文凯[68]则利用最优化问题对粒子群优化算法进行改进，并在 Spark 平台上将改进的粒子群优化算法应用到 K-means 算法中，提高了算法的收敛速度以及运行效率。目前还提出了如动态调整惯性因子 ω [69]、K-means 算法[70]等方法对粒子群优化算法进行了改进。

3.5.3 *其他元启发式算法*

元启发式算法还包括模拟退火算法、蚁群优化算法、人工鱼群算法、人工蜂群算法以及人工神经网络等。

近两年来也有不少学者引入这些算法来提高 K-means 算法的性能，如 El-Shorbagy 等人[71]提出一种混合算法，充分运用遗传算法获得最优解的思想，针对 K-means 聚类过程中的极值点进行突变操作，克服了 K-means 算法的局限性；Garg N 等人[72]将遗传算法与 K-means 结合，利用遗传算法的全局搜索优势确定 K-means 的初始聚类中心，所提出的方法在准确性方面比 K-means 算法要高；文献[73]则在 Spark 平台上采用蝙蝠算法和萤火虫算法等优化算法作为预处理步骤，返回最佳中心点作为 K-means 的输入。

然而，由于这些算法易过早陷入局部最优解，且算法的复杂性使聚类过程耗时较长等缺点，而人工蜂群算法具有收敛速度快、需控制参数少、局部寻优能力强等优点也被应用于改进 K-means 算法中，如 Tran 等人[74]则对原始人工蜂群算法进行改进，在 K-means 每次迭代都利用人工蜂群算法对聚类中心点进行优化，再利用 K-means 算法迭代更新聚类中心点，提高了聚类算法各方面的性能。

除了上述研究，许多新型元启发式算法如灰狼优化算法等也都克服了如易陷入局部最优等缺点，因此，元启发式算法还有很大的发展前景，并且为提高算法的适用性，对每种算法的不同改进方法和算法现今也在不断提出，后续研究还在继续。

4 结束语

近年来，K-means 算法在机器学习领域展现了其特有的优势，受到了广泛关注，但其由于随机初始化聚类中心点，在处理大规模数据时会出现聚类结果不稳定性和算法效率低等现象，是制约算法广泛应用的一大因素。本文总结了解决 K-means 算法聚类结果不稳定现象的研究工作，但 K-means 算法在处理大规模数据时仍面临着巨大的挑战。结合目前的研究现状，未来的研究工作将在以下几个方面展开：

a)现有已改进 K-means 算法优化。现有的改进算法，如本文中介绍的启发式算法和 Canopy 算法等，虽然克服了 K-means 算法的随机初始中心点带来的不稳定等现象，但处理大规模数据时仍由于这些算法的局限性和复杂性，导致运行时间较长，因此，针对当前已有的改进算法，需深入了解它们的原理并分析，例如针对最大最小距离算法优化 K-means 算法时可能会出现聚类冲突现象，新提出最大距离积法等改进方法来避免该现象的发生，进一步提高 K-means 算法的性能。因此，考虑在当前已引入算法的基础上对其缺陷进行改进，并结合 Spark 内存计算框架的特性对算法并行化编程，进一步提高 K-means 算法的时效性，是未来提高 K-means 算法性能的一个重要研究方向。

b)将更多具有良好特性的算法设计优化思想（如元启发式算法人工神经网络算法等）迁移到 K-means 算法初始阶段中。目前改进的 K-means 算法，大多是分阶段进行，即首先需要对数据集预处理从而达到改进 K-means 算法的目的，但这样的改进方法往往由于在实际应用中的适应性低等缺点难以得到广泛应用。解决这个问题的思路是：深入理解其他全局优化方法的核心思想以及针对具体问题处理的模式，再根据 K-means 算法的特点进行调整，从而将其他优化算法应用到 K-means 算法中，不仅可以解决 K-means 算法的聚类效果不佳等问题，并且可能在借鉴过程中产生新的方法或思路。所以，在深入理解现有其他全局优化算法的基础上，如何将其优化思想融入到 K-means 算法中，是未来重要的研究方向之一。

c)面对聚类效果不稳定现象研究新的初始中心点判断方法。当使用 K-means 对数据聚类时，受初始中心点随机选取的影响，无法判定聚类中心点是否会影响聚类结果，即现有的度量方法无法有效判断这一问题。因此，是否可以从聚类结果度量方法入手，研究新的度量方法，在选取初始中心点时，避免中心点选取陷入噪声点中。目前有学者提出如基于余弦距离选取初始中心点的方法，改进了初始中心点的选取方案，从而进一步提高算法性能。然而关于度量方法的提出与证明，既需要充分的数学推理与解释，还需要完整的实验验证等，这也是一个重要的研究方向。

d)基于内存计算框架 Spark 的 K-means 并行化编程。K-means 在对大规模数据聚类时，通常由于其大量的迭代计算无法快速完成，需要借助现有的内存计算框架对算法过程进行加速。所以，对 K-means 算法的改进除了改进算法本身处理过程之外，还需要针对 K-means 算法迭代计算的特点，将其部署适用机器学习迭代计算的 Spark 框架上进行。目前，已有很多学者将改进后的算法结合 Spark 特性并行化编程，为更多的大规模数据应用提供了有效支撑，因此这也是一个具有重要价值的研究方向。

参考文献：

[1] 何清，李宁，罗文娟，等. 大数据下的机器学习算法综述 [J]. 模式识别与人工智能，2014，27(4): 327-336. (He Qing, Li Ning, Luo Wenjuan, *et al.* A survey of machine learning algorithms for big data [J]. Pattern Recognition and Artificial Intelligence, 2014, 27(4): 327-336.)

[2] 贺玲，吴玲达，蔡益朝. 数据挖掘中的聚类算法综述 [J]. 计算机应用研究, 2007, 24(1): 10-13. (He Ling, Wu Lingda, Cai Yichao. Survey of clustering algorithms in data mining [J]. Application Research of Computers, 2007, 24(1): 10-13.)

ChinaXiv合作期刊

[3] Bizer C, Boncz P, Brodie M L, *et al*. The meaningful use of big data: four perspectives-four challenges [J]. ACM SIGMOD Record, 2012, 40 (4): 56-60.

[4] Zaharia M, Chowdhury M, Franklin M J, *et al*. Spark: cluster computing with working sets [C]//Proc of USENIX Conference on Hot Topics in Cloud Computing.Berkeley,CA:USENIX Association, 2010: 10-10.

[5] Zaharia M, Chowdhury M, Das T, *et al*. Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing [C]//Proc of USENIX Conference on Networked Systems Design and Implementation.Berkeley,CA: USENIX Association, 2012: 2.

[6] Shvachko K, Kuang H, Radia S, *et al*. The Hadoop distributed file system[C]//Proc of IEEE Symposium on MASS Storage Systems and Technologies.Washington DC: IEEE Computer Society, 2010: 1-10.

[7] Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters [J]. Communications of the ACM, 2008, 51(1): 107-113.

[8] Lin Xiyqin, Wang Peng, Wu Bin. Log analysis in cloud computing environment with Hadoop and Spark[C]//Proc of IEEE International Conference on Broadband Network & Multimedia Technology. Piscataway,NJ:IEEE Press, 2014: 273-276.

[9] Wan Hu, Xu Yuanchao, Yan Junfeng, *et al*. Mitigating log cost through non-volatile memory and checkpoint optimization [J]. Journal of Computer Research and Development, 2015, 52(6): 1351-1361.

[10] Liao Bin, Yu Jiong, Sun Hua, *et al*. Energy-efficient algorithms for distributed storage system based on block storage structure reconfiguration [J]. Journal of Computer Research & Development, 2015, 48(1): 71-86.

[11] Meng Xiangrui, Bradley J, Yavuz B, *et al*. MLlib: machine learning in Apache Spark [J]. Journal of Machine Learning Research, 2015, 17(1): 1235-1241.

[12] MLlib[DB/OL]. http://spark. apache. org/mllib/

[13] Fatta G D, Pettinger D. Dynamic load balancing in parallel KD-tree k-means [C]//Proc of IEEE International Conference on Computer and Information Technology. Piscataway,NJ:IEEE Press, 2010:2478-2485.

[14] Kusuma I, Ma'Sum M A, Habibie N, *et al*. Design of intelligent K-means based on spark for big data clustering [C]//Proc of International Workshop on Big Data and Information Security. Piscataway,NJ:IEEE Press, 2017: 89-96.

[15] Wang Bowen, Yin Jun, Hua Qi, *et al*. Parallelizing K-means-based clustering on Spark [C]//Proc of International Conference on Advanced Cloud and Big Data. Piscataway,NJ:IEEE Press, 2017: 31-36.

[16] Meng Xiangrui. Spark SQL: Relational data processing in Spark [C]//Proc of ACM SIGMOD International Conference on Management of Data. New York:ACM Press, 2015: 1383-1394.

[17] Xin R S, Rosen J, Zaharia M, *et al*. Shark: SQL and rich analytics at scale [C]//Proc of ACM SIGMOD International Conference on Management of Data. New York:ACM Press, 2013: 13-24.

[18] Gonzalez J E, Xin R S, Dave A, *et al*. GraphX: graph processing in a distributed dataflow framework [C]//Proc of USENIX Conference on Operating Systems Design and Implementation.Berlekey,CA: USENIX Association, 2014: 599-613.

[19] Zaharia M, Das T, Li Haoyuan, *et al*. Discretized streams: fault-tolerant streaming computation at scale [C]//Proc of the 24th ACM Symposium on Operating Systems Principles.2013: 423-438.

[20] Hindman B, Konwinski A, Zaharia M, *et al*. Mesos: a platform for fine-grained resource sharing in the data center [C]//Procs of the 8th USENIX Conference on Networked Systems Design and Implementation. Berlekey,CA: USENIX Association, 2010: 429-483.

[21] Chen Jianguo, Li Kenli, Tang Zhuo, *et al*. A parallel random forest algorithm for big data in a spark cloud computing environment [J]. IEEE Trans on Parallel & Distributed Systems, 2017, 28 (4): 919-933.

[22] 孙吉贵，刘杰，赵连宇. 聚类算法研究 [J]. 软件学报，2008, 19(1): 48-61. (Sun Jigui, Liu Jie, Zhao Lianyu. Clustering Algorithms Research [J]. Journal of Software, 2008, 19(1): 48-61.)

[23] Jain A K, Murty M N, Flynn P J. Data clustering: a review [J]. ACM Computing Surveys, 1999, 31 (3): 264-323.

[24] 行小帅，潘进，焦李成. 基于免疫规划的 K-means 聚类算法 [J]. 计算机学报, 2003,26(5): 605-610. (Xing Xiaoshuai, Pan Jin, Jiao Licheng. A novel K-means clustering based on the immune programming algorithm [J]. Chinese Journal of Computers, 2003,26(5): 605-610.)

[25] Mahajan M, Nimbhorkar P, Varadarajan K. The planar K-means problem is np-hard [C]//Proc of International Workshop on Algorithms and Computation. Berlin:Springer,2009.

[26] Huang Zhexue. A fast clustering algorithm to cluster very large categorical data sets in data mining[C]//Proc of SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery. 1997. 146-151.

[27] 万静，张超，何云斌，等.可变网格优化的 K-means 聚类方法 [J]. 小型微型计算机系统，2018, 39(1): 95-99. (Wan Jing, Zhang Chao, He Yunbin, *et al*. K-means clustering method based on variable grid optimization [J]. Journal of Chinese Computer Systems, 2018, 39(1): 95-99.)

[28] 贾瑞玉，李玉功.类簇数目和初始中心点自确定的 K-means 算法[J]. 计算机工程与应用，2018, 54(7):152-158. (Jia Ruiyu, Li Yugong. K-means algorithm of clustering number and centers self-determination [J]. Computer Engineering and Applications, 2018, 54 (7): 152-158.)

[29] 张波，徐蔚鸿，陈沅涛，等. 基于 Hash 改进的 k-means 算法并行化设计 [J]. 计算机工程与科学, 2016, 38(10): 1980-1985. (Zhang Bo, Xu Weihong, Chen Yuantao, *et al*. A K-means clustering algorithm parallelization design based on Hash [J]. Computer Engineering & Science, 2016, 38(10): 1980-1985.)

[30] 陈小雪，尉永清，任敏，等. 基于萤火虫优化的加权 K-means 算法 [J]. 计算机应用研究，2018, 35(2): 466-470. (Chen Xiaoxue, Wei Yongqing, Ren Min, *et al*. Weighted K-means clustering algorithm based on firefly algorithm [J]. Application Research of Computers, 2018, 35(2): 466-470.)

[31] Duan Yanling, Liu Qun, Xia Shuyin. An improved initialization center k-means clustering algorithm based on distance and density [C]//Proc of American Institute of Physics Conference Series. 2018.

[32] Qian Shenyi, Liu Huihui, Li Daiyi. Research and application of improved K-means algorithm in text clustering [C]//Proc of International Conference on Physics, Computing and Mathematical Modeling. Computer Science and Engieneering. 2018.

[33] Zhu Erzhou, Li Peng, Ma Zhujuan, *et al*. Effective and optimal clustering based on new clustering validity index [C]//Proc of the 22nd IEEE International Conference on Computer Supported Cooperative Work in Design.2018: 529-534.

[34] Nagul S, Kumar R K. An effective K-means approach for imbalance data clustering using precise reduction sampling [J]. International Journal of Computer Sciences and Engineering. 2018, 6(3): 65-70.

[35] Alexandrov A, Bergmann R, Ewen S, *et al*. The Stratosphere platform for big data analytics [J]. The VLDB Journal, 2014, 23(6): 939-964.

chinaXiv:201901.00067v1

ChinaXiv合作期刊

chinaXiv:201901.00067v1

[36] 江小平，李成华，向文，等. K-means 聚类算法的 MapReduce 并行化实现 [J]. 华中科技大学学报:自然科学版，2011，39(s1)：120-124. (Jiang Xiaoping, Li Chenghua, Xiang Wen, *et al.* Parallel implementing K-means clustering algorithm using MapReduce programming mode [J]. Journal of Huazhong University of Science and Technology:Social Science Edition, 2011, 39(s1): 120-124.)

[37] Zhang Dongbo, Shou Yanfang. An improved parallel K-means algorithm based on cloud computing [C]//Proc of International Symposium on Intelligence Computation & Applications. Singapore: Springer, 2015.

[38] Li Chengfei, Tian Guo, Cai KunPeng, *et al.* Improved K-means based on Flink platform and its application in e-commerce big data [C]//Proc of Chinese Automation Congress. 2018.

[39] Kusuma I, Ma'Sum M A, Habibie N, *et al.* Design of intelligent K-means based on spark for big data clustering [C]//Proc of International Workshop on Big Data & Information Security. 2017.

[40] Yuan Tianwei, Xie Xiaolan, *et al.* Design of social security data processing and analysis platform based on Spark [C]// Proc of the 3rd International Conference on Intelligent Information Processing.New York:ACM Press, 2018: 172-177.

[41] Shobanadevi A,Maragatham G. Studying the performance of clustering techniques for biomedical data using spark [C]//Proc of International Conference on Intelligent Sustainable Systems. 2017: 58-65.

[42] Gopalani S, Arora R, Gopalani S, *et al.* Comparing Apache Spark and MapReduce with performance analysis using K-means [J]. International Journal of Computer Applications, 2015, 113(1): 8-11.

[43] Hai Mo, Zhang Yuejing, Li Haifeng, A performance comparison of big data processing platform based on parallel clustering algorithms [J] , Procedia Computer Science, 2018, 139: 127-135.

[44] 陈平华，陈传瑜. 基于满二叉树的二分 K-means 聚类并行推荐算法 [J]. 计算机工程与科学，2015, 37(8): 1450-1457. (Chen Pinghua, Chen Chuanyu. A bisecting K-means clustering parallel recommendation algorithm based on full binary tree [J]. Computer Engineering & Science, 2015, 37(8): 1450-1457.)

[45] 张晓云，刘东. 基于四叉树 K-均值聚类算法的软件故障预测算法研究 [J]. 计算机应用研究，2014, 31(9): 2732-2735. (Zhang Xiaoyun, Liu Dong. Software fault prediction based on quad tree K-means clustering algorithm [J]. Application Research of Computers, 2014, 31 (9): 2732-2735.)

[46] Qi Hui, Di Xiaoqiang, Li Jinqing, *et al.* Improved K-Means Algorithm and Its Application to Vehicle Steering Identification [C]//Proc of International Conference on Advanced Hybrid Information Processing. Cham: Springer, 2017.

[47] Bentley J L. Multidimensional divide-and-conquer [J]. Communications of the ACM, 1980, 23(4): 214-229.

[48] Tiwari S, Solanki T, Tiwari S, *et al.* An optimized approach for K-means clustering [C]// Foundation of Computer Science, 2013: 5-7.

[49] Redmond S J, Heneghan C. A method for initialising the K-means clustering algorithm using kd-trees [J]. Pattern Recognition Letters, 2007, 28(8): 965-973.

[50] Kumar K M, Reddy A R M. An efficient K-means clustering filtering algorithm using density based initial cluster centers [J]. Information Sciences, 2017, 418-419.

[51] 马菁，李力. RDD上扩展索引层优化的分布式K-means算法 [J//OL]. 计算机工程与应用.[2018-11-22]. http://kns. cnki. net/kcms/detail/11. 2127. TP. 20180515. 1558. 002. html. (Ma Jing, Li Li. Optimization of distributed K-means algorithm with RDD-based extended index layer [J]. Computer Engineering and Applications. [2018-11-22].)

[52] Mccallum A. Efficient clustering of high-dimensional data sets with application to reference matching [C]//Proc of International Conference on Knowledge Discovery and Data Mining. 2000: 169-178.

[53] Zhang Geng, Zhang Chengchang, Zhang Huayu. Improved K-means algorithm based on density Canopy [J]. Knowledge-Based Systems, 2018, 145.

[54] 徐鹏程，王诚. K-means 算法改进及基于 Spark 计算模型的实现[J]. 南京邮电大学学报:自然科学版,2017, 37(4): 113-118. (Xu Pengcheng, Wang Cheng. Improvement of K-means algorithm and implementation based on Spark computing model [J]. Journal of Nanjing University of Posts and Telecommunications:Natural Science Edition, 2017, 37(4): 113-118.)

[55] Katsavounidis I, Jay Kuo C C, Zhang Z. A new initialization technique for generalized Lloyd iteration [J]. IEEE Signal Processing Letters, 1994, 1 (10): 144-146.

[56] Tzortzis G, Likas A. The MinMax K-means clustering algorithm [J]. Pattern Recognition, 2014, 47(7): 2505-2516.

[57] 毛典辉. 基于 MapReduce 的 Canopy-Kmeans 改进算法 [J]. 计算机工程与应用，2012, 48(27): 22-26,68. (Mao Dianhui. Improved Canopy-Kmeans algorithm based on MapReduce [J]. Computer Engineering & Applications, 2012, 48 (27): 22-26,68.)

[58] Tang K S, Man K F, Kwong S, *et al.* Genetic algorithms and their applications [J]. IEEE Signal Processing Magazine, 1996, 13 (6): 22-37.

[59] Glover F, Marti R. Tabu Search [J]. General Information, 1997, 106(2): 221-225.

[60] Gao Hao, Pun Chiman, Kwong S. An efficient image segmentation method based on a hybrid particle swarm algorithm with learning strategy [J]. Information Sciences, 2016, 369: 500-521.

[61] Kennedy J, Eberhart R. Particle swarm optimization [C]//Proc of IEEE International Conference on Neural Networks, 2002: 1942-1948.

[62] Cao Buyang, Glover F, Rego C. A Tabu search algorithm for cohesive clustering problems [J]. Journal of Heuristics, 2015, 21(4): 457-477.

[63] Xu Zheng, Cao Buyang. A parallel Tabu search algorithm with solution space partition for cohesive clustering problems [M]// Algorithms and Architectures for Parallel Processing. Springer International Publishing, 2015: 333-343.

[64] Lu Yinhao, Cao Buyang, Rego C, *et al.* A Tabu search based clustering algorithm and its parallel implementation on Spark [J]. Applied Soft Computing, 2017, 63: 97-109.

[65] 杨志，罗可. 一种改进的基于粒子群的聚类算法 [J]. 计算机应用研究，2014, 31(9): 2597-2599. (Yang Zhi, Luo Ke. Improved clustering algorithm based on particle swarm optimization [J]. Application Research of Computers, 2014, 31(9): 2597-2599.)

[66] Lin Youcheng, Tong Nan, Shi Majie, *et al.* K-means optimization clustering algorithm based on particle swarm optimization and multiclass merging [J]. Computer Systems & Applications, 2014, 168: 569-578.

[67] 许明杰，蔚承建，沈航. 基于Spark的并行K-means算法研究 [J]. 微电子学与计算机，2018, 35(5): 95-99. (Xu Mingjie, Wei Chengjian, Shen Hang. Research on K-means algorithm of Spark parallelization [J]. Microelectronics & Computer, 2018, 35(5): 95-99.)

[68] 刘文凯. 改进的粒子群算法及其在聚类算法中的应用 [D]. 广州：广东工业大学，2017. (Liu Wenkai. Improved particle swarm optimization algorithm and its application in clustering algorithm [D]. Guangzhou:

ChinaXiv合作期刊

Guangdong University of Technology, 2017.)

[69] 李立军，张晓光. 基于动态粒子群优化与 K-means 聚类的图像分割算法[J]. 现代电子技术，2018，41(10):164-168. (Li Lijun，Zhang Xiaoguang. Image segmentation algorithm based on dynamic particle swarm optimization and K-means clustering [J]. Modern Electronics Technique, 2018, 41(10): 164-168.)

[70] 班俊硕，赖惠成，林宪峰，等. 改进 PSO 与 K-均值聚类肤色分割的人脸检测算法 [J]. 激光杂志，2017，38(2): 82-86. (Ban Junshuo, Lai Huicheng, Lin Xianfeng, *et al.* Face detection method with improved PSO and K-means clustering based on skin color segmentation [J]. Laser Journal, 2017, 38(2): 82-86.)

[71] El-Shorbagy M A, Ayoub A Y, El-Desoky I M, *et al.* A novel genetic algorithm based K-means algorithm for cluster analysis[C]// Proc of International Conference on Advanced Machine Learning Technologies and Applications. Cham:Springer, 2018: 92-101.

[72] Garg N, Gupta R K. Performance evaluation of new text mining method based on GA and K-means clustering algorithm [J]. Advances in Intelligent Systems and Computing,2018, 562: 23-30.

[73] Santhi V, Jose R. Performance analysis of parallel K-means with optimization algorithms for clustering on Spark [C]//Proc of International Conference on Distributed Computing and Internet Technology. Berlin: Springer, 2018: 158-162.

[74] Tran Dangcong, Wu Zhijian, Wang Zelin, et al. A novel hybrid data clustering algorithm based on artificial bee colony algorithm and K-means [J]. Chinese Journal of Electronics, 2015, 24(4): 694-701.

chinaXiv:201901.00067v1